POSTER: Example Based Motion Generation with Efficient Control

Yujean Choung
Soongsil University
MAGIC Lab., IT building 602
Sangdo-dong, Dongjak-gu
156-743, Seoul, Korea
yjchoung@magiclab.kr

Kyung-Kyu Kang
Soongsil University
MAGIC Lab., IT building 602
Sangdo-dong, Dongjak-gu
156-743, Seoul, Korea
rcrookie@magiclab.kr

Dongho Kim
Soongsil University
MAGIC Lab., IT building 602
Sangdo-dong, Dongjak-gu
156-743, Seoul, Korea
prof@magiclab.kr

ABSTRACT

This paper presents a motion generation technique for arbitrary morphologies with the user defined correspondences between joints. Users can define the controlling part in the source character and the part to be controlled in the target character in our system. To remove the restriction in the morphology of the target character, we use the pair of example posture sets. In our system, in order to provide the correspondence regardless of the number of joints, the deformed part in the target character is simplified into the direction vector. The final postures are then generated with the weighted sum of the examples. Our experimental results demonstrate that our approach can generate motions for various target characters and can control the user defined joints. Exploiting this correspondence between joints, our system can reflect the intention of animators efficiently and reduce their manual efforts.

Keywords

Character animation, Motion generation, Motion blending, Motion retargeting, Motion synthesis

1. INTRODUCTION

Motion capture technology has been used in movies and animations for long time. Many motion retargeting algorithms have been developed to apply the motions to 3D virtual characters. Most of the retargeting researches can generate target characters which have the same structure as the source character but different only in the segment lengths [Gle98, SLSG01]. Recently, there has been growing interest in the motion generation research which supports different skeletal structures between the source and target characters. Kulpa et al. [KMA05] proposed a method that can apply motions without limitation in the number of bones by defining a new human skeletal structure using the links between the joints of the body. To remove the restriction in the morphology, there have been two kinds of approaches. One is an illustrative way that creates the motion data expressed in novel contexts [HRE+08], and the other way is the example based approach [PS04].

The example based methods usually generate the final postures through the weighted sum of the given set of examples [PS04, LLC+07]. Park and Shin [PS04] presented Motion cloning method. It uses the pair of examples where source and target characters are related one by one. They compute the weights for the source character examples which have the identical structure to the input character. Then they generate the motion of target character by applying weights for the corresponding pairs. So this method does not have any limitation in morphologies of the targets. But they need an additional pair of examples for the whole body even if users want to create a posture for only some part of the character. This may result in the motions which the users did not expect.

This paper uses the pair of examples because our system is based on Motion cloning technique to set the target character's morphologies without limitation in the number of bones. But our system enables the users to control parts of the character freely with existing examples. We aimed to reflect the intention of the animator without additional manual efforts.

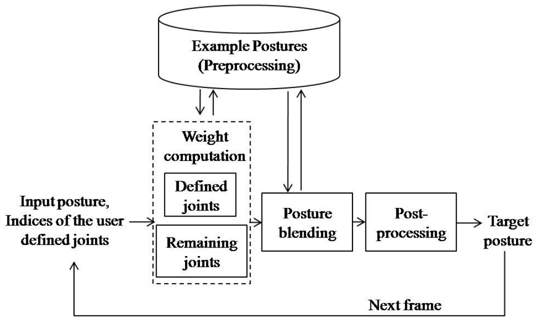


Figure 1. Overview of the motion generation system

2. Overview

Motion generation system

Figure 1 illustrates our system. Our motion generation system is similar to the motion synthesis process in the motion cloning method. But we developed the process for the user defined correspondence between joints. The system generates the posture of the target character corresponding to the input posture per frame in real-time. It is generated by the weighted sum of example postures.

The example postures are prepared in the preprocessing stage. Let P_i^s and P_i^t ($1 \le i \le n_{example}$) denote the set of source example postures and that of their corresponding target example postures, where i is the index of the examples and $n_{example}$ is the number of pairs.

The input data are the posture in each frame of the given motion and the joint indices defined by the user. The system calculates two sets of weights, one is for the user defined joints and the other one is for the remaining joints. Then we blend the postures P_i^t using those weights, and finally make an output through the post-processing.

User defined correspondence between joints

The user can define the controlling part in the source character and the part to be controlled in the target character. Figure 2 illustrates the example of correspondence between joints. s_s and s_e denote the starting and the end point of the controlling part. t_s, t_e denote the starting and the end point to be controlled.

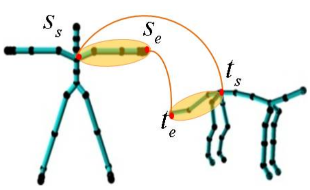


Figure 2. The example of user defined correspondence between joints

Those points are expressed by joint indices. Corresponding target joints can be controlled by the input motion. So the system generates both the whole postures and the controlled part of the target character at the same time.

3. Motion synthesis

Linearization of the Postures

A posture can be denoted by $P = (p, q_1, q_2, ..., q_{n_{joint}})^T$ where $p \in \mathbb{R}^3$ is the position and $q \in \mathbb{S}^3$ is the orientation of each joint. Because of the non-linearity of the unit quaternion space, it is hard to calculate the weights which are used to blend the examples. So we used posture linearization process [PSS02, PS04]. All the orientations are converted into their corresponding displacement vectors using the logarithm map and they can be recovered using the exponential map. So the final linearized posture can be represented by $\hat{P} = (0, 0, v_2, ..., v_{n_{joint}})^T$, $(1 \le j \le n_{joint})$ where j is the joint index and n_{joint} is the total number of joints.

Weight computation for the remaining joints

In this process, the system computes the weight for the remaining joints excluding the user defined joints. The posture of the remaining joints are determined by the weight w_i^g of P_i^s, so that P_i^s matches P^{input} as much as possible. In the following equation, $\hat{P}^{input}$ and $\hat{P}_i^s$ are linearized input posture and the example postures respectively.

$$\hat{P}^{input} = \sum_{i=1}^{n_{example}} w_i^g \hat{P}_i^s , \qquad (1)$$

We used damped least square method to solve this linear system for w_i^g.

Then the weights are applied to the corresponding example postures to generate the final target character posture.

Weight computation for the user defined joints

In this process, the system computes the weight w_i^p for the corresponded joints $t_s \le j \le t_e$. The motion cloning method [PS04] needs to be provided with the new example postures of the whole joints when the user wants to make the new correspondence between joints. But our system only needs the existing examples using direction vectors between the joints. The direction vector removes the restriction in the number of joints and simplifies the high degree of freedom [KMA05]. So we can make the postures without additional examples.

 ISBN 978-80-86943-95-4

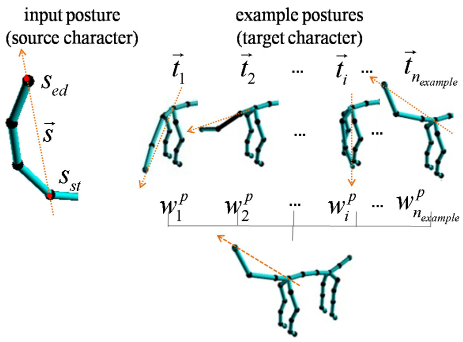


Figure 3. The weight computation for defined joints

We calculate the direction vector $\vec{s}$ toward the joint s_e from the joint s_s of the input posture P^{input}. The system uses the vector $\vec{t}_i$ toward the joint t_e from the joint t_s of the example postures P_i^t of the target character. The system restores the $\vec{s}$ direction with blending $\vec{t}_i$ using weights w_i^p. This process can be expressed as follows:

$$\vec{s} = \sum_{i=1}^{n_{example}} w_i^p \vec{t}_i \,. \tag{2}$$

The weights w_i^p are calculated by solving the linear system. The linear system for calculating weights is expressed as follows:

$$\begin{bmatrix} \vec{t}_{x,1} & \vec{t}_{x,2} & \cdots & \vec{t}_{x,n_{example}} \\ \vec{t}_{y,1} & \vec{t}_{y,2} & \cdots & \vec{t}_{y,n_{example}} \\ \vec{t}_{z,1} & \vec{t}_{z,2} & \cdots & \vec{t}_{z,n_{example}} \end{bmatrix} \begin{bmatrix} w_1^p \\ w_2^p \\ \vdots \\ w_{n_{example}}^p \end{bmatrix} = \begin{bmatrix} \vec{s}_x \\ \vec{s}_y \\ \vec{s}_z \end{bmatrix} . \tag{3}$$

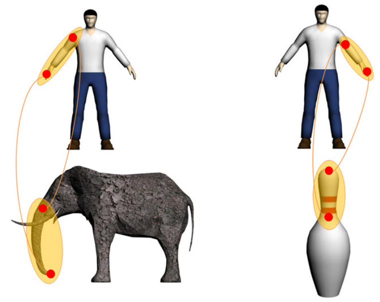
The rows of the matrix on the left hand side represent x, y, z values of $\vec{t}_i$, and columns are examples. We adopt the damped least square method to solve this linear system. This method generates the shape of the target character according to the source character motions. With the examples and the directional vector, plausible postures can be generated even if the numbers of bones are different.

Posture blending and post-processing

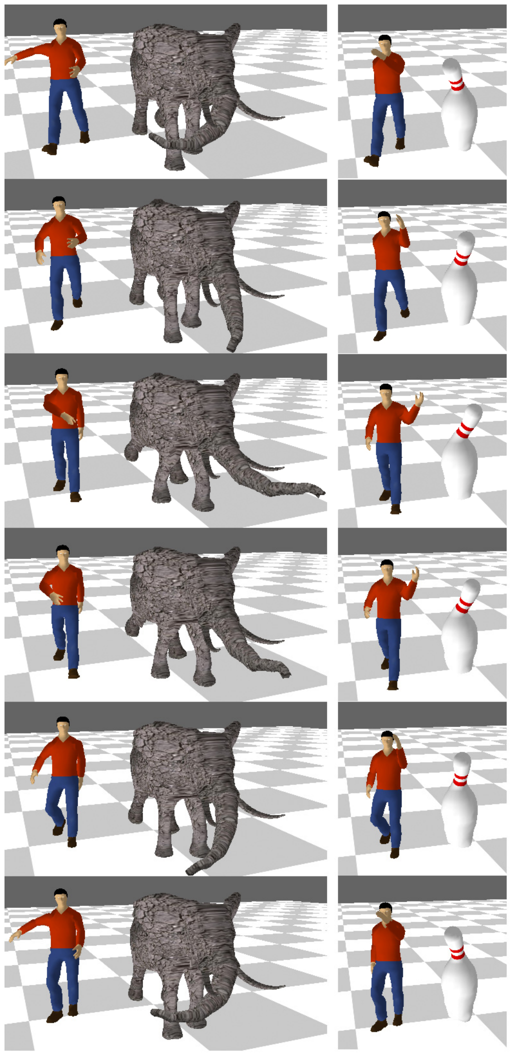
The system finally generates the target character's posture P^{output} with applying w_i^g and w_i^p to P_i^t. The example posture P_i^t is linearized to $\hat{P}_i^t$ and the weights are applied as follows:

$$\hat{P}^{output} = \sum_{i=1}^{n_{example}} w_i \hat{P}_i^t \,. \tag{4}$$

In the above equation, w_i is determined by the following rule:



(a)

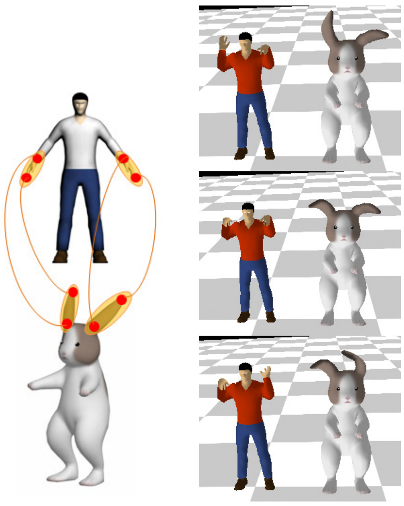


(b)

Figure 4. Elephant and bowling pin:
(a) The correspondence between joints (b) The results of our method

$$w_i = \begin{cases} w_i^p & if\ t_s \le j \le t_e, \\ w_i^g & otherwise \end{cases} \,. \tag{5}$$

 ISBN 978-80-86943-95-4



(a) (b)

Figure 5. Rabbit: (a) The correspondence between joints (b) The results of our method

In the post-processing step, root position of the target character is constructed by scaling the trajectory of the input motion and adapting to the target character.

4. EXPERIMENTAL RESULTS

The examples in this paper were computed on a 1.86GHz Intel Core2Duo CPU, with 2GB RAM and nVIDIA geForce 8800GTS graphics card. We used the source character which has human-like structures with 26 bones. Figure 4 shows the results after generating the elephant and the bowling pin's locomotion. The elephant has 36 bones and different articulated structure from the source character. In the result (a), the user corresponds the right arm to the elephant's nose. With 11 pairs of examples, the elephant locomotion and nose poses were generated. Bowling pin has 14 bones and also different articulated structure from the source character. The user corresponds the left arm from elbow to finger to the bowling pin from neck to head. It used 15 pairs of examples. Figure 5 shows the result of rabbit example with two corresponding parts.

5. CONCLUSION

In this article, we have presented a motion generation technique for arbitrary skeletal structures with the user defined correspondences between joints. By simplifying the controlled part to the direction vector, the system can generate the motion without additional examples and does not restrict the number of bones. So it can reduce the animator's manual efforts. Exploiting this correspondence between joints, our system can generate the part of the posture according to the intention of the animator, which is hard to generate previously.

If the existing examples don't have enough movements, the generated part may not meet the user's intention. We are planning to solve this problem as future research. The interactivity of our system and no limitation in the structure of target character could be applied to many other applications such as games or interactive art works.

6. ACKNOWLEDGMENTS

This work was supported by grant No. R01-2006-000-11214-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

7. REFERENCES

[BB04] Baerlocher, P., and Boulie, R. An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels. The Visual Computer, 20(6), pp.402-417, 2004.

[Gle98] Gleicher, M. Retargetting Motion to New Character. 25th Computer graphics & Interactive techniques SIGGRAPH'98, pp.33-42, 1998.

[HRE+08] Hecker, C., Raabe, B., Enslow, R.W., DeWeese, J., Maynard. J., and Prooijen. K.v. Real-time Motion Retargeting to Highly Varied User-Created Morphologies. ACM Transaction on Graphics 2008, 27(3), pp.27:1-27:11, 2008.

[KMA05] Kulpa, R., Multon, F., and Arnaldi, B. Morphology independent representation of motions for interactive human-like animation. Computer Graphics Forum, Eurographics 2005 special issue 24(33), pp.343-352, 2005.

[LLC+07] Lee, T.Y., Lin, C.H., Chu, H.K., Wang, Y.S., Yen, S.W., and Tsai, C.R. Mesh pose-editing using examples. International Conference on Computer Animation and Social Agents, 18(4-5), pp.235-245, 2007.

[PS04] Park, M.J., and Shin, S.Y. Example-based motion cloning. Computer Animation and Virtual World, 15, pp.245-257, 2004.

[PSS02] Park, S.I., Shin, H.J., and Shin, S.Y. On-line Locomotion Generation Based on Motion Blending. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp.105-111, 2002.

[SLSG01] Shin, H.J., Lee, J., Shin, S.Y., and Gleicher, M. Computer puppetry: an importance-based approach. ACM Transactions on Graphics 2001, 20(2), pp.67-94, 2001.